# FP-Tree and Its Variants: Towards Solving the Pattern Mining Challenges

**Anindita Borah and Bhabesh Nath**

**Abstract** Mining patterns from databases is like searching for precious gems which is a gruesome task but still a rewarding one. The frequent patterns are believed to be valuable assets for the researchers that provide them useful information. The frequent and rare pattern mining paradigm is broadly divided into Apriori and FP-Tree-based approaches. Experimental results and performance evaluation available in the literature have established the fact that FP-Tree-based approaches are superior to the Apriori ones on various grounds. This paper explores the various modifications of FP-Tree that were developed to tackle the major pattern mining research challenges. Through this paper, an attempt has been made to review the usefulness and applicability of the most eminent data structure in the domain of pattern mining, the FP-Tree.

**Keywords** Frequent patterns · FP-Tree · Pattern mining · Challenges

## 1 Introduction

Pattern mining has established itself as a significant field of data mining over the years. The notion of frequent pattern mining emphasizes that useful information may be hidden among the frequently occurring patterns in a database. Since its inception, there were several attempts from the frequent pattern mining researchers for extracting such interesting patterns. The significance of frequent patterns was

A. Borah (✉) · B. Nath
Department of Computer Science & Engineering, Tezpur University,
Napaam, Sonitpur 784028, Assam, India
e-mail: anindita01.borah@gmail.com

B. Nath
e-mail: bnath@tezu.ernet.in



A.K. Somani et al. (eds.), *Proceedings of First International Conference on Smart System, Innovations and Computing*, Smart Innovation, Systems and Technologies 79, https://doi.org/10.1007/978-981-10-5828-8_51

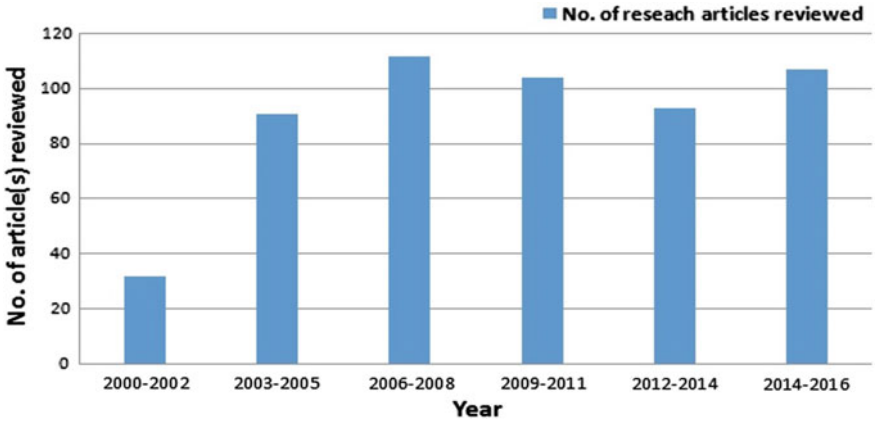


**Fig. 1** Number of articles published based on FP-Tree

first identified in [1]. They developed a level-wise approach named Apriori for mining the frequent patterns by generating candidates. This primer approach proved to be inefficient as it generates an enormous quantity of candidate patterns as well as performs multiple scanning of the database. With a view to overcome the shortcomings of Apriori method, Han et al. [2] proposed a data structure called Frequent Pattern Tree (FP-Tree) that retains the complete database information. Initially, it collects the item frequencies present in the database and then generates the conditional bases and conditional FP-Trees from which the frequent patterns are extracted.

Performance evaluation illustrates that FP-Growth is the most efficient frequent pattern mining method and performs better than Apriori on various grounds. A year-wise distribution of the FP-Tree-based articles published is shown in Fig. 1. Only relevant and recent articles have been considered, excluding review articles or general surveys. The graph in the figure indicates that a commendable amount of research based on FP-Tree has been carried out over the years. Considering the significance and popularity of the FP-Tree data structure, many of its variants were proposed and various pattern mining algorithms employed the same for handling the research issues. An overview of all those attempts is therefore necessary to let the researchers get introduced with the usefulness of FP-Tree.

The remaining paper, followed by the introduction is systematized as follows: Sect. 2 discusses the various research issues that were resolved using FP-Tree based approaches. To illustrate the efficiency of FP-Tree-based approaches, a comparative analysis of the same with Apriori and its variants is provided in Sect. 3. Finally some future prospects for FP-Tree-based approaches and conclusion is discussed in Sect. 4.

## 2 Research Issues Handled by FP-Tree-Based Approaches

Since its outset, FP-Tree-based approaches have attempted to solve various research challenges confronted by the pattern mining community. Due to its importance and significance, the usability of FP-Tree is widespread. This section discusses the different endeavors made by FP-Tree-based approaches to handle the major pattern mining research challenges.

### *2.1 Improvement in Efficiency*

In spite of being one of the widely accepted pattern mining technique, there are many instances where the efficiency of FP-Growth demands improvement. Several variations of FP-Tree have been proposed over the years that attempted to enhance the effectiveness of FP-Growth algorithm. Liu et al. [3] tried to lessen the search space of FP-Growth by employing ascending frequency order to construct the prefix trees as well as for search exploration. With a view to upgrade the effectiveness of FP-Growth algorithm on grounds of space and execution time, Racz [4] developed an alternative data structure that is more condensed than FP-Tree and also allows faster traversal and allocation.

One of the major drawbacks of FP-Growth is its imperative demand for memory. TD-FP-Growth developed by [5], avoids the generation of conditional pattern bases as well as conditional FP-Trees to reduce the amount of memory consumption. Sucahyo and Gopalan [6] symbolize the transaction items in main memory using Compressed FP-Tree (CFP-Tree) in order to lessen the memory usage. FP-Growth algorithm is also inefficient in handling sparse datasets. FPGrowth algorithm developed by Grahne and Zhu [7] attempts to solve this issue using an array based implementation.

### *2.2 Mining Frequent Closed Itemsets*

Setting up a low support threshold might generate an enormous number of frequent itemsets in the database. It has been found that instead of looking for each and every frequent itemset in the database, it is better to obtain a significant class of frequent itemset called *frequent closed itemset*. A *frequent closed itemset* is one, all of whose proper subsets are frequent. Pei et al. [8] developed the CLOSET algorithm with the purpose of generating the *frequent closed itemsets*. For fast exploration of *frequent closed itemsets*, a modified compression technique called CLOSET+ [9] is used based on single-prefix path. Another algorithm called FPClose [10], employs a tree data structure called CFI-Tree to check how close the frequent itemsets are.

### 2.3 Secondary Memory Based

Limitation of main memory is one of the serious bottlenecks of pattern mining techniques while mining large databases. Adnan and Alhajj [11] recognized this major issue and developed a secondary memory-based approach for mining the frequent patterns. Their proposed data structure behaves similar to FP-Tree upon construction in main memory but gets modified to a disk-resident structure only when the FP-Tree can no longer be accommodated in physical memory. A similar approach was adopted by Bonchi and Goethals [12] to cater to the needs of main memory.

### 2.4 Handling Incremental Datasets

A major challenge encountered by the pattern mining techniques is to deal with dynamic or incremental datasets. It is worthless starting the entire mining process from scratch if any update occurs in the database. Incremental Frequent Pattern Growth (IFP-Growth) [13] is an enhanced version of the traditional FP-Growth algorithm that handles the addition or deletion of data in dynamic databases. A similar strategy is adopted by the AFPIM algorithm [14]. Cheung and Zaiane [15] developed CATS Tree that executes only a single scan of the database and gives better results in terms of storage compression. CanTree [16] and CP-Tree [17] are other single-pass algorithms for incremental mining of the database.

### 2.5 Mining Frequent Patterns from Uncertain Data

Mining patterns from uncertain data have established itself as an emerging field of research over the years. Leung et al. [18] proposed a new technique called UF-Growth in order to generate patterns from uncertain data. Calders et al. [19] proposed a variation of FP-Growth called UFP-Growth that focuses on extracting frequent patterns from uncertain data. Their UFP-Tree data structure stores the probabilistic information of the items in each node and computes the expected support of every item during the first scan.

### 2.6 Mining High Utility Itemsets

Identifying high utility or profitable itemsets from databases is an indispensable task of data mining. High utility itemsets signify those itemsets that have high importance or are more profitable for the users. To generate the high utility itemsets,

Tseng et al. [20] developed a FP-Growth-based approach called Utility Pattern Growth (UP-Growth) where the information about the high utility itemsets is stored in a tree structure called Utility Pattern Tree (UP-Tree) which is further extended in [21]. Lin et al. [22] exploited the property of FP-Tree in their proposed data structure called High Utility Pattern Tree (HUP-Tree). Their approach integrates the strategies of two-phase algorithm.

## 2.7 Mining Sequential Patterns

Sequential pattern mining has established itself as an important data mining application over the years. Lin et al. [23] proposed a sequential pattern mining data structure called Fast Updated Sequential Pattern Tree (FUSP-Tree). In order to represent the sequence relation between two connecting nodes, the link between them is imprinted with the symbol *s* whereas to represent the relation between items, the link is imprinted with the symbol *i*. Pei et al. [24] proposed another FP-Tree-based data structure called Web Access Pattern Tree (WAP-Tree) for mining frequent sequential patterns from web logs. The data structure holds information about access patterns and the mining algorithm then generates the access patterns from log.

## 2.8 Mining Maximal Frequent Itemsets

Tremendous number of itemset generation has always been the major issue encountered by frequent pattern mining techniques. To lessen the number of candidate subsets formed, some of the existing pattern mining techniques emphasize the generation of *maximal frequent itemsets*. In case of *maximal frequent itemsets*, none of the superset is frequent. Grahne and Zhu [25] developed a recursive algorithm called FPMax to mine the MFI's where a linked list is used to store the items of the conditional pattern bases during the current call. Yan et al. [26] developed the Frequent Pattern Tree for Maximal Frequent Itemsets (FPMFI) that employs a projection based superset checking technique to find out the MFI's.

## 2.9 Handling Data Streams

Frequent pattern mining over data streams comes up with diverse challenges. The rapid and continuous flow of data stream makes the mining and update of frequent patterns a difficult task. Giannella et al. [27] proposed a variation of FP-Growth algorithm called FP-stream with a view to generate the frequent patterns from data streams. The algorithm is based on a *tilted time window* framework and employs

two tree data structures. DS-Tree data structure developed by Leung et al. [28], stores the information of the data streams in a canonical order. Tanbeer et al. [29] proposed a data structure CPS-Tree that employs a sliding window strategy and partitions the window into a series of transactions called *pane*.

### *2.10 Mining Rare Patterns*

Recent studies show that in several domains, the rare items are of greater interest as compared to the frequent ones. This insisted on mining and retaining rare items that are removed during the frequent itemset generation phase. Hu and Chen [30] modified the FP-Growth algorithm by incorporating multiple minimum supports during itemset generation. Another variant of FP-Growth was proposed by Tsang et al. [31] called the Rare Pattern Tree (RP-Tree) algorithm that generates only the rare itemsets, discarding the frequent ones. Bhatt and Patel [32] further extended the RP-tree algorithm using the maximum constraint model to improve its efficiency.

### *2.11 Handling Big Data*

Extracting frequent patterns from big data using traditional pattern mining techniques is a difficult and problematic task. To handle big data, Chen et al. [33] developed the Parallel FP-Growth (PFP-Growth) algorithm to allow parallel processing of big data. Leung and Hayduk [34] developed the MR-Growth algorithm that generates frequent patterns from big uncertain data. The concepts of FP-Growth and MapReduce are combined in this novel method to extract frequent patterns from large quantities of uncertain data. To incrementally update the frequent itemsets in big data, Chang et al. [35] proposed a method that employs a heap data structure and outperforms existing algorithms in terms of complexity as well as execution time.

## 3 Comparison with Apriori and Its Variants

The FP-Tree and Apriori-based approaches have been extensively used for handling the pattern mining challenges. This section presents a general comparison of the research work carried out, employing these two standard strategies. The graphical analysis given in Fig. 2, illustrates a comparison between the number of approaches developed under Apriori and FP-Growth. The blue bar in the graph represents FP-Tree-based approaches while the red bar represents Apriori-based approaches respectively. From the graph it can be observed that except two issues, that is extracting rare patterns and frequent pattern mining from

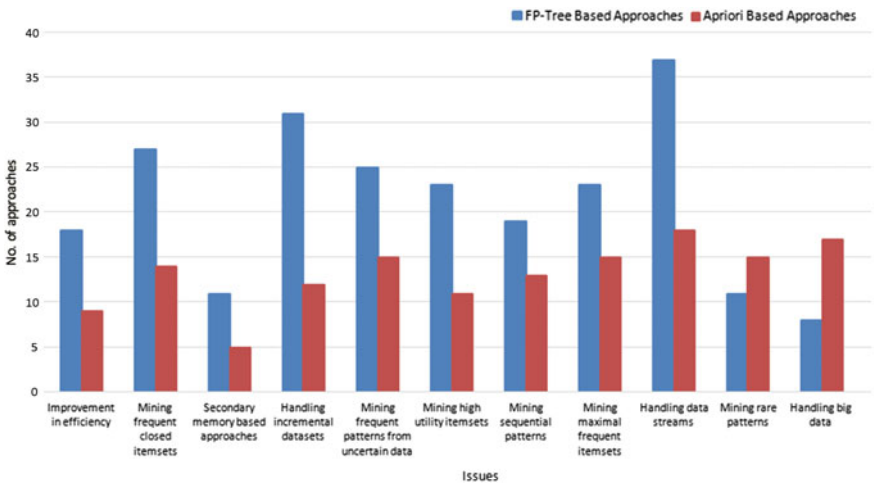


**Fig. 2** Comparison between Apriori and FP-Tree-based approaches

big data, FP-Tree-based approaches are extensively explored over Apriori-based approaches. This clearly establishes the superiority of FP-Tree-based approaches over the Apriori ones.

## 4 Conclusion and Future Prospects

The pattern growth approaches have been found to be more efficient among the pattern mining techniques. Among the pattern growth approaches, the most prominent and favorable one is the FP-Growth algorithm. From Sect. 2, the relevance of FP-Tree based approaches can be recognized. The comparative study given in Sect. 3, establishes the superiority of FP-Tree-based approaches over Apriori ones. However, there are still some issues that are not pervasively addressed by the FP-Tree-based approaches.

FP-Tree based approaches are inefficient in handling sparse datasets. In spite of this fact, only one attempt has been made to improve its efficiency in this regard. Rare pattern mining being a new and emerging area, have not explored the FP-Tree based approaches to much extent. Only a limited number of approaches based on FP-Tree can be found in the literature. Another less explored issue by FP-Tree-based approaches is mining patterns from big data. The approaches based on FP-Tree are quite less than Apriori approaches for handling this issue. Considering the popularity and effectiveness of FP-Tree-based approaches, the researchers need to work on the less explored issues to contribute some fruitful

pattern mining techniques to the research community. Even though, literature has endowed plentiful FP-Tree-based approaches to the pattern mining community, there is still much room for expansion.

## References


1. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: Acm sigmod record. vol. 22, pp. 207–216. ACM (1993).
2. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM Sigmod Record. vol. 29, pp. 1–12. ACM (2000).
3. Liu, G., Lu, H., Yu, J.X., Wang, W., Xiao, X.: Afopt: An efficient implementation of pattern growth approach. In: FIMI (2003).
4. Racz, B.: nonordfp: An fp-growth variation without rebuilding the fp-tree. In: FIMI (2004).
5. Wang, K., Tang, L., Han, J., Liu, J.: Top down fp-growth for association rule mining. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 334–340. Springer (2002).
6. Sucahyo, Y.G., Gopalan, R.P.: Ct-pro: A bottom-up non recursive frequent Itemset mining algorithm using compressed fp-tree data structure. In: FIMI. vol. 4, pp. 212–223 (2004).
7. Grahne, G., Zhu, J.: Efficiently using prefix-trees in mining frequent itemsets. In: FIMI. vol. 90 (2003).
8. Pei, J., Han, J., Mao, R., et al.: Closet: An efficient algorithm for mining frequent closed itemsets. In: ACM SIGMOD workshop on research issues in data mining and knowledge discovery. vol. 4, pp. 21–30 (2000).
9. Wang, J., Han, J., Pei, J.: Closet+: Searching for the best strategies for mining frequent closed itemsets. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 236–245. ACM (2003).
10. Grahne, G., Zhu, J.: Fast algorithms for frequent itemset mining using fp-trees. IEEE transactions on knowledge and data engineering 17(10), 1347–1362 (2005).
11. Adnan, M., Alhajj, R.: Drfp-tree: disk-resident frequent pattern tree. Applied Intelligence 30 (2), 84–97 (2009).
12. Bonchi, F., Goethals, B.: Fp-bonsai: the art of growing and pruning small fp-trees. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 155–160. Springer (2004).
13. Xu, B., Yi, T., Wu, F., Chen, Z.: An incremental updating algorithm for mining association rules. Journal of Electronics (China) 19(4), 403–407 (2002).
14. Koh, J.L., Shieh, S.F.: An efficient approach for maintaining association rules based on adjusting fp-tree structures. In: International Conference on Database Systems for Advanced Applications. pp. 417–424. Springer (2004).
15. Cheung, W., Zaiane, O.R.: Incremental mining of frequent patterns without candidate generation or support constraint. In: Database Engineering and Applications Symposium, 2003. Proceedings. Seventh International. pp. 111–116. IEEE (2003).
16. Leung, C.K.S., Khan, Q.I., Li, Z., Hoque, T.: Cantree: a canonical-order tree for incremental frequent-pattern mining. Knowledge and Information Systems 11(3), 287–311 (2007).
17. Tanbeer, S.K., Ahmed, C.F., Jeong, B.S., Lee, Y.K.: Cp-tree: a tree structure for single-pass frequent pattern mining. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 1022–1027. Springer (2008).
18. Leung, C.K.S., Carmichael, C.L., Hao, B.: Efficient mining of frequent patterns from uncertain data. In: Seventh IEEE International Conference on Data Mining Workshops (ICDMW 2007). pp. 489–494. IEEE (2007).

19. Calders, T., Garboni, C., Goethals, B.: Efficient pattern mining of uncertain data with sampling. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 480–487. Springer (2010).
20. Tseng, V.S., Wu, C.W., Shie, B.E., Yu, P.S.: Up-growth: an efficient algorithm for high utility itemset mining. In: Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 253–262. ACM (2010).
21. Tseng, V.S., Shie, B.E., Wu, C.W., Philip, S.Y.: Efficient algorithms for mining high utility itemsets from transactional databases. IEEE transactions on knowledge and data engineering 25(8), 1772–1786 (2013).
22. Lin, C.W., Hong, T.P., Lu, W.H.: An effective tree structure for mining high utility itemsets. Expert Systems with Applications 38(6), 7419–7424 (2011).
23. Lin, C.W., Hong, T.P., Lu, W.H., Lin, W.Y.: An incremental fusp-tree maintenance algorithm. In: 2008 Eighth International Conference on Intelligent Systems Design and Applications. vol. 1, pp. 445–449. IEEE (2008).
24. Pei, J., Han, J., Mortazavi-Asl, B., Zhu, H.: Mining access patterns efficiently from web logs. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 396–407. Springer (2000).
25. Grahne, G., Zhu, J.: High performance mining of maximal frequent itemsets. In:6th International Workshop on High Performance Data Mining (2003).
26. Yan, Y.J., Li, Z.J., Chen, H.W.: Efficiently mining of maximal frequent item sets based on fp-tree. Ruan Jian Xue Bao (J. Softw.) 16(2), 215–222 (2005).
27. Giannella, C., Han, J., Pei, J., Yan, X., Yu, P.S.: Mining frequent patterns in data streams at multiple time granularities. Next generation data mining 212, 191–212 (2003).
28. Leung, C.K.S., Khan, Q.I.: Dstree: a tree structure for the mining of frequent sets from data streams. In: Sixth International Conference on Data Mining (ICDM'06). pp. 928–932. IEEE (2006).
29. Tanbeer, S.K., Ahmed, C.F., Jeong, B.S., Lee, Y.K.: Efficient frequent pattern mining over data streams. In: Proceedings of the 17th ACM conference on Information and knowledge management. pp. 1447–1448. ACM (2008).
30. Hu, Y.H., Chen, Y.L.: Mining association rules with multiple minimum supports: anew mining algorithm and a support tuning mechanism. Decision Support Systems 42(1), 1–24 (2006).
31. Tsang, S., Koh, Y.S., Dobbie, G.: Rp-tree: rare pattern tree mining. In: Data Warehousing and Knowledge Discovery, pp. 277–288. Springer (2011).
32. Bhatt, U., Patel, P.: A novel approach for finding rare items based on multiple minimum support framework. Procedia Computer Science 57, 1088–1095 (2015).
33. Chen, M., Gao, X., Li, H.: An efficient parallel fp-growth algorithm. In: Cyber-Enabled Distributed Computing and Knowledge Discovery, 2009. CyberC'09. International Conference on. pp. 283–286. IEEE (2009).
34. Leung, C.K.S., Hayduk, Y.: Mining frequent patterns from uncertain data with map reduce for big data analytics. In: International Conference on Database Systems for Advanced Applications. pp. 440–455. Springer (2013).
35. Chang, H.Y., Lin, J.C., Cheng, M.L., Huang, S.C.: A novel incremental data mining algorithm based on fp-growth for big data. In: Networking and Network Applications (NaNA), 2016 International Conference on. pp. 375–378. IEEE (2016).